

01-27-00

A

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application Transmittal

Assistant Commissioner for Patents
Washington, D.C. 20231

10/25 U.S. PTO
09/491582
01/25/00

Sir:

Transmitted herewith for filing is the Patent Application of:

Inventor: T. Inagaki et al.For: METHOD AND DEVICE TO PROCESS MULTIDIMENSIONAL ARRAY OBJECTS

Enclosed are:

- ☒ 6 sheets of drawings
☒ A Declaration and Power of Attorney (with missing signatures)
☐ An Information Disclosure Statement and form PTO-1449
☒ A certified copy of a Japanese application, Application No. 11-017943, filed January 27, 1999
☐ An assignment of the invention to International Business Machines Corporation, Armonk, New York 10504 and Recordation form PTO-1595
☐

The filing fee has been calculated as follows:

For:	No. Filed	No. Extra
Basic Fee		
Total Claims	21 -20 =	1
Indep. Claims	3 -3 =	0
<input type="checkbox"/> Multiple Dependent Claim Presented		

Other Than Small Entity

Rate	Fee
	\$ 690.00
x \$18.00=	18.00
x \$78.00=	0.00
\$260.00	\$ 0.00
TOTAL	708.00

EXPRESS MAIL CERTIFICATE

Express Mail Label No.: EK489516111US
 Date: January 25, 2000

I hereby certify that I am depositing the enclosed or attached paper with the U.S. Postal Service "Express Mail Post Office to Addressee" service on the above date, addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Owen J. Gamon
 Owen J. Gamon


Patent Application Transmittal
 Attorney Docket No.: JA998-218

Deposit Account Authorization:

- ☒ Please charge Deposit Account No. 09-0465 in the amount of \$708.00. A duplicate copy of this sheet is enclosed.
- ☒ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account 09-0465. A duplicate copy of this sheet is enclosed.
- ☒ Any additional filing fees required under 37 C.F.R. §1.16.
- ☒ Any patent application processing fees under 37 C.F.R. §1.17.

Respectfully submitted,

By


Owen J. Gamon
Registration No.: 36,143

Date: January 25, 2000

IBM Corporation
Intellectual Property Law, Dept. 917
3605 Highway 52 North
Rochester, MN 55901-7829

(507) 253-4660 voice
(507) 253-2382 fax

METHOD AND DEVICE TO PROCESS MULTIDIMENSIONAL ARRAY OBJECTS

Field of the Invention

5 The present invention relates to a method and a device for processing a multidimensional array object in a language in which a multidimensional array is implemented by an array of array objects.

Background

Processing speed can be dramatically improved by paralleling and vectorizing a process for elements of a multidimensional array. In particular, in a processor with multiple execution units such as IA64 or PowerPC, an effect of improved processing speed by such optimizations is remarkable. In order to perform such optimizations, it is necessary to analyze dependency among processes of multidimensional array elements and locations of array elements. However, in a language like Java which implements a multidimensional array by means of an array of array objects, it is difficult to perform such analyses since element configuration of a multidimensional array may change at an execution time.

20 Conventionally, there were roughly two solutions to this problem. First, there is a known method of performing an analysis by scanning addresses of multidimensional array elements accessed by a certain process at an execution time. There is also a known method of implementing a
25 multidimensional array without any change in element configuration at an execution time by means of extension of language specifications or installation of a unique library. It certainly becomes possible to optimize a process for a multidimensional array by using these methods.

However, there were following various problems in these methods.

First, as to the former method, in addition to taking too much time for scanning addresses, there was a problem of having to stop a garbage collector from the time the address scanning is started until processing of an array ends. In particular, the problem of the time for address scanning is fatal for Java in which processing by Just-in-Time compiler is becoming general, so it is not a realistic solution.

On the other hand, as to the latter method, it is proposed by Sun Microsystems Inc. to use a class to access, as a multidimensional array, an area to which memory is allocated as a one-dimensional array. It first provides a class as follows.

```
final class FloatMatrix2D {
    private final float m;
    private final int cols;
    public FloatMatrix2D(int c, int r)
    {
        cols=c;
        m=new float c*r ;
    }
    public float get(int c, int r)
    {
        return m cols*r+c ;
    }
}
```

Here, it is a method wherein, by using method in-lining and operator overloading, an expression described as,

```
FloatMatrix2D f=new FloatMatrix2D(cols, rows)
```

```

f i1,j1 =f i2, j2 +4;
is handled as follows,
f i1,j1 =f i2, j2 +4;

```

```

5  T=f.get(i2,j2)+4;
   f.put(i1,j1,T);

```

```

T=f cols*j2+i2 +4;
f cols*j1+i1 =T;

```

10 By using such a method, it is possible to directly access elements without undergoing reference to any object array. In addition, it is not necessary to check an index as required when referring to an object array. In this method, however, there is a problem that application of an analysis required for parallelization or optimization becomes difficult.

Namely, in the above method, as there is no information in the source rendered into bytecode, it is not possible to perform any more analysis of dependency among array accesses than an analysis in an index converted to one dimension.

20 Accordingly, it becomes very difficult to detect parallelism taking multidimensionality into consideration, so application of parallelization cannot be expected in many cases.

For instance,

```
for j
```

```
25   for i
```

```
       f i,j =f 2*i-1,j ;
```

in a program as above, since it can easily be analyzed as having no dependency in the direction of j, great parallelism can be obtained by skewing the innermost loop in the direction of j or unrolling the loop in the direction of j. However, if the bytecode is converted to $f \text{ cols} * j + i = f \text{ cols} * j + 2 * i - 1$ it becomes necessary to seek if there is an integer solution in

cols*j1+i1=cols*j2+2*i2-1

and decide whether there is any dependency as to the direction of j . This means that a simple analytic problem has become an NP complete problem. Also in general, it is thinkable that a problem becomes more difficult to solve if multidimensional information reduces to one dimension, and thus the method of Sun Microsystems Inc. is not a method suited to dependency analysis for parallelization.

There is also another method to allow dependency analysis by modifying or extending language specifications. For instance, a method as in Fortran wherein a multidimensional array is introduced to a language is thinkable. Or even in the above-mentioned method wherein an array is simply one-dimensionalized, dependency analysis should be made possible by adding extended specifications as follows. Namely, in the following expression used in the preceding example,

$f \text{ cols}^*j+i = f \text{ cols}^*j+2*i-1$

if language specifications are extended so that a compiler may take advantage of the array being two-dimensional and $f.cols$ being the number of columns, comparison of indexes can be multidimensionally developed so as to facilitate an analysis. For instance, in this example, the problem can be replaced with the following,

an analysis of $(cols^*j+i) \text{div cols}$
and $(cols^*j+2*i-1) \text{div cols}$
an analysis of $(cols^*j+i) \text{mod cols}$
and $(cols^*j+2*i-1) \text{mod cols}$

so it can easily be analyzed as having parallelism in the direction of j . However, costs for such modifications and extensions of specifications are very high. With respect to Java in particular, in consideration of the present

circumstances where Java is spreading worldwide, it is not easy to effectively extend its specifications by addition of a new array object or by dependence on a specific library.

5 In addition, while a method to analyze an expression tree of an arithmetic expression without extending language specifications is also thinkable, it has a problem that an analysis becomes difficult when the expression tree is transformed due to deletion of a common subexpression and so on in addition to requirement of support to specify the dimensionality.

Summary

15 An object of the present invention is to resolve the above-mentioned subject and provide a method and a device for processing a multidimensional array object which allows improved speed of processing a multidimensional array without any modification of specifications.

20 The present invention covers a method for processing a multidimensional array object in a language in which a multidimensional array is implemented by an array of array objects. And to a multidimensional array object comprising array objects which constitute the multidimensional array, flags representing that it is possible to optimize a process for elements of the multidimensional array object are added as additional information. The flags are stored in a storage device (main memory for instance). Then, a machine code corresponding to a state of the flags is executed.

The present invention introduces flags representing that it is possible to optimize a process for elements of a multidimensional array object and monitors modifications of element configuration of a multidimensional array, and at the same time, assures that a process optimized for an array to which these flags are set may be executed. Thus, it becomes possible, without any modification of specifications, to determine at high speed a possibility of optimizing a process for a multidimensional array and process a multidimensional array whose process optimization is determined to be possible with an optimized code so as to improve execution speed of a program. Consequently, it becomes possible to speed up a process of a multidimensional array implemented by an array of array objects, which was conventionally difficult.

Brief Description of the Drawings

Fig. 1 is a drawing for explaining an example of multidimensional array objects in the present invention.

Fig. 2 is a drawing for explaining an example of pointers in multidimensional array objects of the present invention.

Fig. 3 is a drawing for explaining an example of handling of flags representing that it is possible to optimize a process at a writing time in multidimensional array objects of the present invention.

Fig. 4 is an overview of the entire system of the present invention.

Fig. 5 is a flowchart showing an example of processes at a compilation time in the present invention.

Fig. 6 is a flowchart showing an example of processes when generating a multidimensional array in the present invention.

Fig. 7 is a flowchart showing an example of processes when writing to array objects which constitute a multidimensional array in the present invention.

Fig. 8 is a flowchart showing an example of processes when accessing a multidimensional array in the present invention.

Detailed Description

In a language like Java covered by the present invention, a multidimensional array is implemented by an array of array objects. As contents of such an object array (an array of array objects) may be freely rewritten, there is a possibility that element configuration of a multidimensional array may change at an execution time. Accordingly, it is very difficult to determine whether it is possible to optimize a process for each element of a multidimensional array. On the other hand, in reality, many multidimensional arrays are generated with element configuration which allows optimization, and it seldom happens that a possibility of optimizing the process is destroyed by rewriting of an object array. Thus, in many cases, there are circumstances where it is possible to optimize the process of a multidimensional array but optimization cannot be performed. In the present invention, on the above precondition, flags representing that it is possible to optimize a process for elements of a multidimensional array object are added as additional

information to a multidimensional array object comprising array objects which constitute the multidimensional array, and executes a machine code corresponding to a state of the flags.

As a preferred embodiment, a machine code corresponding to a state of the flags may either be dynamically generated at an execution time or provided in advance. Moreover, the flags are configured so as to be inverted if a predetermined condition is no longer met. This predetermined condition means, for instance, that a base array of a multidimensional array object is allocated to consecutive memory areas. In addition, as a preferred embodiment, a machine code of a portion for processing a multidimensional array object is either a machine code optimized or a machine code not optimized according to the predetermined condition corresponding to a state of the flags. Furthermore, it is determined whether or not the predetermined condition is met when writing to the multidimensional array object so that the flags are configured to be inverted if the predetermined condition is not met. On the other hand, if the predetermined condition is met when generating the multidimensional array object, it is configured so that the flags are set to a generated multidimensional array object.

As a probability in a language like Java, in the case that there is a possibility of multi-thread processing of a multidimensional array object, a machine code for storing on a stack a dummy reference to the multidimensional array during execution of an optimization code is generated so as to prevent collection by a garbage collector.

Next, an example of implementing the method for processing a multidimensional array object of the present

invention in Java Just-in-Time compiler is explained.

First, device configuration of the present invention in an environment using Java is described by using Fig. 4. Server computer 1 and client computer 5 are connected via network 3. Client computer 5 comprises Java VM (virtual machine) 52, OS (operating system) 53 and hardware (including CPU and memory) 55. Moreover, Java VM 52 comprises Java interpreter 54 or Java JIT compiler 56. It may also comprise both interpreter 54 and JIT compiler 56. Meanwhile, client computer 5 may be, other than an ordinary computer, a so-called network computer or a home information appliance which has smaller size of memory or does not include any auxiliary storage such as a hard disk.

On server computer 1, Java source code 10 is compiled by Java compiler 12. The result of this compilation is bytecode 14. This bytecode 14 is sent to client computer 5 via network 3. Bytecode 14 is a native code for Java Virtual Machine (Java VM) 52 installed on a WWW browser (World Wide Web Browser) in client computer 5, etc., and Java interpreter 54 or Java JIT compiler 56 is used when actually executed on the CPU of hardware 55. Interpreter 54 decodes bytecode 14 at an execution time, and invokes and executes a processing routine prepared for each instruction. On the other hand, JIT compiler 56 converts a bytecode to machine code 58 by using a compiler in advance or immediately before execution and then executes it on the CPU.

JIT compiler 56 implements a method for processing a multidimensional array object described in detail as follows. Here, a condition to allow optimization of a process for elements of a multidimensional array object is that an array

of the lowest dimension in which elements of a multidimensional array object are stored (hereafter, a base array) is allocated to consecutive memory areas. Fig. 1 shows an example of it. Placement of elements of a multidimensional array object is kept in such a state so that a dependency analysis among processes accessing the elements becomes possible. In this embodiment, a multidimensional array is generated so that a base array may be consecutively placed when multianewarray instruction of Java bytecode is executed, and thereafter, implementation is performed to drop the flags when writing occurs to array objects which constitute the multidimensional array (see Fig. 1) by astore instruction. The concrete process regarding the above is shown in 1.1 to 1.2. In addition, since a process is executed in multi-thread in Java, special measures taking it into consideration are required. This is referred to in 1.3. While 1.2 describes a procedure for processing, a machine code corresponding to the processing is generated for implementation. Lastly, reduction of execution time when a process is added to astore is discussed in 1.4.

1.1: Processing at a compilation time (Fig. 5)

The condition to allow optimization of a process for elements of a multidimensional array object is that a base array of a multidimensional array is allocated to consecutive memory areas.

1. On the precondition that the base array is allocated to consecutive memory areas, a machine code with an optimized process is generated.

2. A machine code not optimized is generated, which processes elements of a multidimensional array object.

1.2: Processing at an execution time (Fig. 6)

1.2.1: Processing when generating a multidimensional array

The following process is performed when multianewarray instruction is executed.

1. Calculate the memory size necessary for all elements of a multidimensional array object.

2. Allocate memory of the calculated size.

3. Generate a base array to the allocated areas.

4. Calculate the memory size necessary for an array of array objects which constitute the multidimensional array.

5. Allocate memory of the calculated size.

6. Create in the allocated areas an array of array objects which constitute the multidimensional array.

7. Set the flags representing that it is possible to optimize a process of the array created in 6.

8. Add to the array created in 6 a pointer to an array of array objects of higher order. Fig. 2 shows an example of it.

1.2.2: Processing when writing to array objects which constitute a multidimensional array (Fig. 7)

The following process is performed when aastore instruction is executed.

1. Drop the flags representing that it is possible to optimize a process of the array to be written to.

2. Recursively drop the flags of an array of higher order with reference to a pointer to a higher order array of the array to be written to. Leave the flags as is as to an array of lower order whose continuity of a base array is maintained. Fig. 3 shows an example of it.

3. After the above updating of the flags ends, contents of the array objects are modified.

1.2.3: Processing when accessing a multidimensional array (Fig. 8)

1. Read the flags representing that it is possible to optimize a process of the array to be processed.

2. If the flags are set to the subject to be processed, have an optimization code executed.

3. If the flags are not set to the subject to be processed and there exists any lower dimension in the array to be processed, recursively perform this process to the lower dimension.

4. If the flags are not set to the subject to be processed and there is no lower dimension in the array to be processed, have a non-optimization code executed.

1.2.3.1: Concrete example

The following shows a concrete example of a multidimensional array access.

....

xA 100 100 =1.0;

....

for i= 0 to 100

5 for K= 0 to 100

 vB i =vB i +xA k i *vB i

In this example, as xA 100 100 is accessed preceding a loop, there is essentially no exception occurring to the access to xA in the loop. However, as there is a possibility of rewriting a multidimensional array by astore, in such an example, it is necessary to check an index range for every access to xA k i in the innermost loop unless all xA 0, xA 1, ..., xA 100 are privatized (copied to local of a thread). On the other hand, use of the present invention improves this code as follows.

....

xA 100 100 =1.0;

....

for i= 0 to 100

20 if(IsOptimizable(xA)) {

 for k= 0 to 100

 vB i =vB i +xA k i *vB i ;//Generate a code without
any range check

 } else {

25 for k= 0 to 100

 vB i =vB i +xA k i *vB i ;// Generate a code with a
range check
 }

30 The above optimization eliminates conditional branching
from the innermost loop and significantly reduces execution
time.

1.3: Support for multi-thread processing

In a multi-thread environment, there is a possibility that multiple processes perform reading/writing in parallel to the flags representing that it is possible to optimize a process of the same multidimensional array. For a system using the present invention to properly operate, support in case such an access occurs becomes necessary. The following shows concrete measures.

1.3.1: Assurance of consistency between a flagged array and an actually accessed array

If an address storing array elements is acquired and flags are read in separate cycles, there is a possibility that another process may rewrite these values between the two cycles. In this case, since contents of flags representing that it is possible to optimize a process and an array for which a possibility of optimizing a process is designated by the flags do not match, a system utilizing the present invention may violate the language specifications of Java. This problem can generally be solved by rendering these scans into critical sections, while here, a nature property to this embodiment is used and the problem is solved in the following method.

1. When entering in an optimization routine, reading is performed in order of element addresses of an accessed array to flags. In this embodiment, flags change only once from a state of being on to a state of not being on. Accordingly, if flags are on at a reading time, it can be assured that contents of the flags and what is designated by them certainly match. Moreover, if flags are not on, no problem arises since any code accessing the array is a code not optimized.

(A concrete example)

....


```

for i= 0 to 100
  if(IsOptimizable(xA)) {
    /*Optimization routine*/
    for k = 0 to 100
      vB i =vB i +xA k i *vB i ;//No range check
    } else
      ....

```

For instance, in the aforementioned innermost loop, a machine language wherein xA accesses are all performed by using relative addresses from a starting address of a base array is generated. If reading is performed in order of flags to a starting address, in the case that an interrupt by aastore (such as xA 1 =null;) occurs during the two readings, the language specifications of Java is violated. On the other hand, if reading is performed in order of an address to flags, there is not such a problem.

2. Writing is performed in order of flags to element addresses of an accessed array. If scanning is performed in this order, the language specifications is not violated even when an optimization code starts operating during writing of the flags and address changes.

3. In a multidimensional array which is three-dimensional or more, writing of flags is hierarchically performed. Here, in order to retain the language specifications of Java, writing of flags is performed in order of higher to lower dimensions.

(A concrete example)

For instance, if flags are dropped in order of lower to higher dimensions, an optimization code of a higher dimension

may be started during this process. If this code is one that does not check flags of an array of a lower dimension (such as the above code), there is a danger that the language specifications of Java may be violated.

1.3.2: Control of a garbage collector

If a process A is calculating element addresses for a multidimensional array possible to optimize a process by utilizing continuity of a base array and not using an object array, there is a possibility that another process B may modify an array of array objects which constitute the multidimensional array during the process of the array. In the specifications of Java, if these operations are invoked in an asynchronous method, process A may perform a process assuming continuity of a base array. However, there is a possibility that a part of the base array is referred to from nowhere due to change of an object array by process B and the array elements to be processed are collected by a garbage collector during processing of process A. To cope with this problem, in this embodiment, collection by a garbage collector is prevented by generating a machine code for storing on a stack a dummy reference to the array during processing by an optimization code.

1.4: Discussion of a side effect caused by a process added to aastore.

An example of a program

```
public void sortObj(Obj[]obj){
    for(int i=0;i<obj.length;i++)
        for(int j=i;j<obj.length;j++)
            if(obj[i].data >= obj[j].data){
                Obj tmp = obj[i];
                obj[i] = obj[j];
                obj[j] = tmp;
            }
}
```

```

        obj[j] = tmp;
    }
}

```

Bytecode for the innermost loop

```

5  Label1:
    aload 1 <obj[]>
    iload 2 <i>
    aaload <obj[i]>
    getfield <dat>
10  aload 1 <obj[]>
    iload 3 <j>
    aaload <obj[j]>
    getfield <dat>
    if_icmplt Label2:
15  aload 1 <obj[]>
    iload 2 <i>
    aaload <obj[i]>
    astore 4 <tmp>
    aload 1 <obj[]>
20  iload 2 <i>
    aload 1 <obj[]>
    iload 3 <j>
    aaload <obj[j]>
    astore <obj[i] = obj[j]>
25  aload 1 <obj>
    iload 3 <j>
    aload 4 <tmp>
    astore <obj[j] = tmp>
    Label2:
30  iinc 3 1 <j++>
    aload 1 <obj>
    arraylength <obj.length>

```

if_icmplt Label1:

Since flags representing that it is possible to optimize a process cannot be set when this array is generated, it is also possible to determine flags outside a loop and perform a versioning for a loop with contents of these flags. Meanwhile, "with a side effect" in this case means to implement a process related to the present invention at an execution time of astore instruction, and "without a side effect" means only to execute simple astore instruction.

```

10 if( IsNotOptimizable(obj) ){
    for(int i=0;i<obj.length;i++){
        for(int j=i;j<obj.length;j++){
            if(obj[i].data >= obj[j].data){
                Obj tmp = obj[i];
                obj[i] = obj[j]; //generate astore without a side
                                effect
                obj[j] = tmp;    //generate astore without a side
                                effect
            }
        }
    }
} else {
    for(int i=0;i<obj.length;i++){
        for(int j=i;j<obj.length;j++){
25     if(obj[i].data >= obj[j].data){
            Obj tmp = obj[i];
            obj[i] = obj[j]; //generate astore with a side
                            effect
            obj[j] = tmp;    //generate astore with a side
                            effect
30     }
    }
}

```

```
}  
}  
}
```

5 It should be noted that the condition of the first if sentence
is IsNotOptimizable(obj). Namely, in the case of a program
which cannot be optimized, it is not necessary to implement
any process related to the present invention for aastore,
while in the case that it can be optimized, a process related
to the present invention is implemented for aastore (from the
10 else sentence on).

As it is clear from the above discussion, the present
invention introduces flags representing that it is possible to
optimize a process for elements of a multidimensional array
object and monitors modifications of element configuration of
a multidimensional array, and at the same time, assures that a
code optimized for an array to which these flags are set may
be executed, and thus, it becomes possible, without any
modification of specifications, to determine at high speed a
possibility of optimizing a process for a multidimensional
array and process a multidimensional array whose process
optimization is determined to be possible with an optimized
code so as to improve execution speed of a program.
Consequently, it becomes possible to speed up a process of a
multidimensional array implemented by an array of array
25 objects in Java and so on, which was conventionally difficult.

What is claimed is:

CLAIMS

1 1. A method for processing a multidimensional array object
2 comprising array objects, said method comprising the steps of:

3 managing flags for said multidimensional array object,
4 said flags representing whether it is possible to optimize a
5 process for elements of said multidimensional array object;
6 and
7 executing a machine code corresponding to a state of said
8 flags.

9 2. The method of claim 1, further comprising:

10 inverting said flags when a predetermined condition is no
11 longer met.

12 3. The method of claim 2, wherein said predetermined
13 condition is whether a base array of a multidimensional array
14 object is allocated to consecutive memory areas.

15 4. The method of claim 2, wherein said machine code is either
16 a machine code optimized or a machine code not optimized
17 according to said predetermined condition.

18 5. The method of claim 2, further comprising:
19 determining whether said predetermined condition is met
20 when writing to said multidimensional array object.

21 6. The method of claim 2 wherein, further comprising:
22 if said predetermined condition is met when generating
23 said multidimensional array object, setting said flags to a
24 generated multidimensional array object.

1 7. The method of claim 1 wherein, further comprising:
2 if there is possibility of multi-thread processing of
3 said multidimensional array object, generating a machine code
4 for storing on a stack a dummy reference to said
5 multidimensional array during execution of an optimization
6 code.

1 8. A storage medium storing a program for a multidimensional
2 array object comprising array objects, wherein said program,
3 when read and executed by a computer, comprises steps of:
4 managing flags for said multidimensional array object,
5 said flags representing that it is possible to optimize a
6 process for elements of said multidimensional array object;
and
executing a machine code corresponding to a state of said
flags.

9. The storage medium of claim 8, further comprising:

inverting said flags when a predetermined condition is no
longer met.

1 10. The storage medium of claim 9, wherein said predetermined
2 condition is whether a base array of a multidimensional array
3 object is allocated to consecutive memory areas.

1 11. The storage medium of claim 9, wherein said machine code
2 is either a machine code optimized or a machine code not
3 optimized according to said predetermined condition.

1 12. The storage medium of claim 9, further comprising:
2 determining whether said predetermined condition is met
3 when writing to said multidimensional array object.

1 13. The storage medium of claim 9, further comprising:
2 if said predetermined condition is met when generating
3 said multidimensional array object, setting said flags to a
4 generated multidimensional array object.

1 14. The storage medium of claim 8 wherein, further
2 comprising:
3 if there is possibility of multi-thread processing of
4 said multidimensional array object, generating a machine code
5 for storing on a stack a dummy reference to said
6 multidimensional array during execution of an optimization
7 code.

1 15. A computer for processing a multidimensional array object
2 comprising array objects, said computer comprising:
3 a central processing unit; and
4 a program, when read and executed by said central
5 processing unit, comprises steps of:
6 managing flags for said multidimensional array object,
7 said flags representing that it is possible to optimize a
8 process for elements of said multidimensional array object,
9 and
10 executing a machine code corresponding to a state of said
11 flags.

1 16. The computer of claim 15, wherein said program further
2 comprises:

3 inverting said flags when a predetermined condition is no
4 longer met.

1 17. The computer of claim 16, wherein said predetermined
2 condition is whether a base array of a multidimensional array
3 object is allocated to consecutive memory areas.

1 18. The computer of claim 16, wherein said machine code is
2 either a machine code optimized or a machine code not
3 optimized according to said predetermined condition.

1 19. The computer of claim 16, wherein said program further
2 comprises:
3 determining whether said predetermined condition is met
4 when writing to said multidimensional array object.

1 20. The computer of claim 16, wherein said program further
2 comprises:
3 if said predetermined condition is met when generating
4 said multidimensional array object, setting said flags to a
5 generated multidimensional array object.

1 21. The computer of claim 15 wherein, said program further
2 comprises:
3 if there is possibility of multi-thread processing of
4 said multidimensional array object, generating a machine code
5 for storing on a stack a dummy reference to said
6 multidimensional array during execution of an optimization
7 code.

METHOD AND DEVICE TO PROCESS MULTIDIMENSIONAL ARRAY OBJECTS

Abstract of the Disclosure

5 A method for processing a multidimensional array object in which a multidimensional array is implemented by an array of array objects. The multidimensional array object comprises array objects which constitute the multidimensional array. Flags representing that it is possible to optimize a process for elements of the multidimensional array object are added as additional information. The flags are stored in a storage device (main memory for instance). Then, a machine code corresponding to a state of the flags is executed.

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209

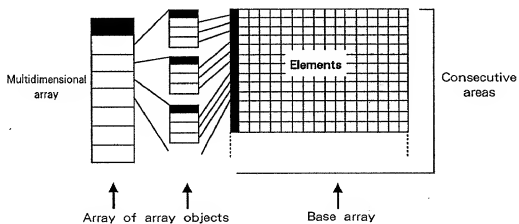


Fig. 1

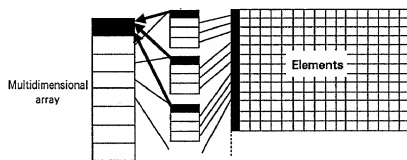


Fig. 2

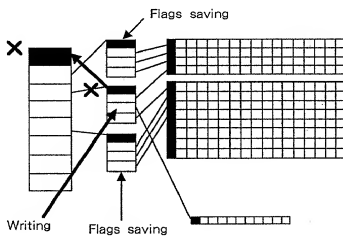


Fig. 3

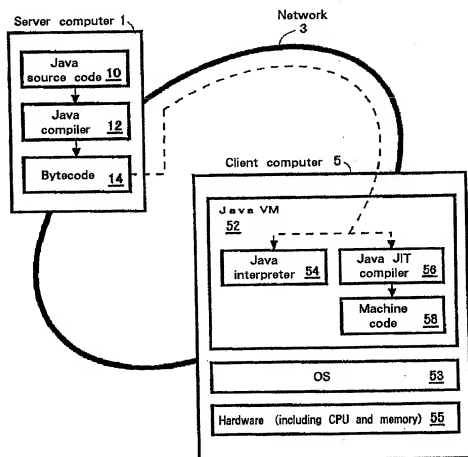


Fig. 4

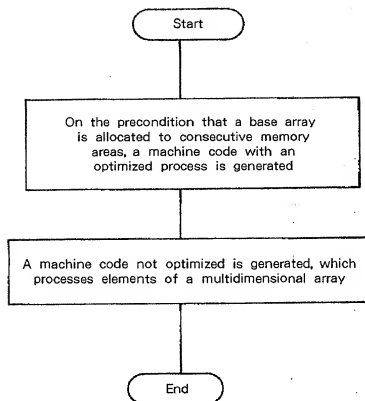


Fig. 5

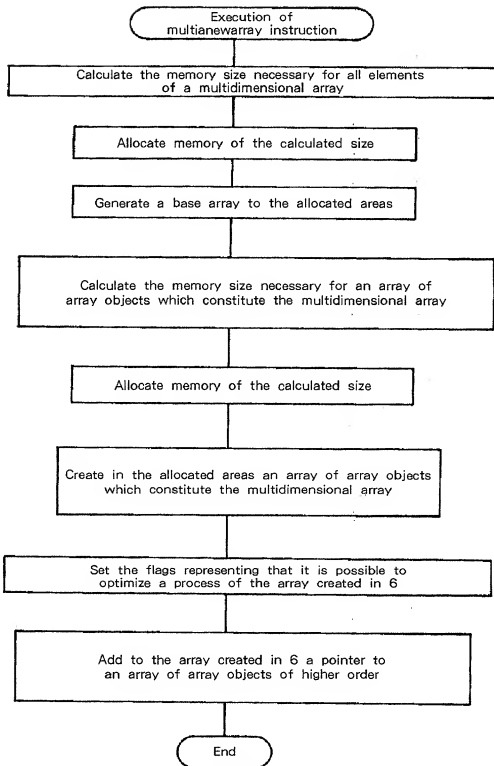


Fig. 6

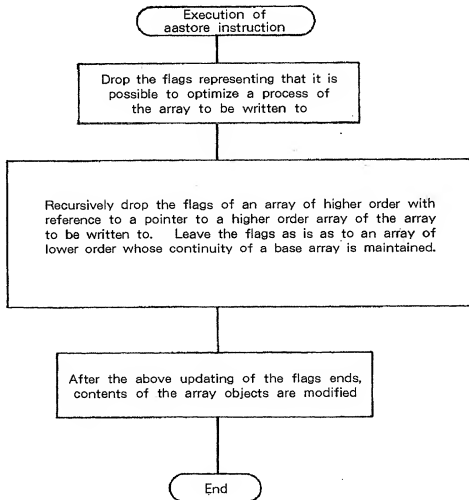


Fig. 7

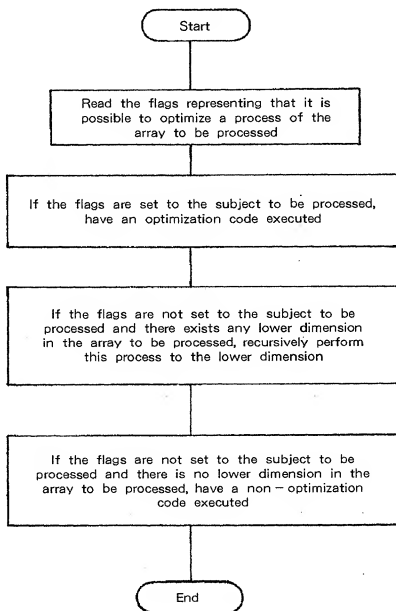


Fig. 8

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION
Docket No.: JA998-218

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name. I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND DEVICE TO PROCESS MULTIDIMENSIONAL
ARRAY OBJECTS

the specification of which (check one)

X is attached hereto.

_____ was filed on _____ as
Application Serial No. _____ and
was amended on _____

(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

Priority Claimed

11-017943 Japan 27/01/1999 X YES NO
(Number) (Country) (Day/Month/Year Filed)

I hereby claim the benefit under Title 35, United States Code, §120 of any United States Application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

04900582-012500

_____(NONE)_____
(Application Serial No.) (Filing Date) (Status) (Patented, Pending, Abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.
(List name and registration number)

Matthew J. Bussan - 33,614	John E. Hoel - 26,279
Owen J. Gamon - 36,143	Christopher A. Hughes - 26,914
Pryor A. Garnett - 32,136	Edward A. Pennington - 32,588
James R. Nock - 42,937	Joseph C. Redmond, Jr. - 18,753
Steven W. Roth - 34,712	

Send Correspondence to: Owen J. Gamon
IBM Corporation, Dept. 917
3605 Highway 52 North
Rochester, MN 55901-7829

Direct Telephone Calls to: Owen J. Gamon
Area Code 507-253-4660

Full name of sole or first Inventor

Tatsushi Inagaki
Inventor's signature Date

Residence 75-10-202, Kanamori, Machida-shi, Tokyo-to
Citizenship Japan
Post Office Address
Same as above

Full name of second Inventor

Hideaki Komatsu
Inventor's signature

Date

Residence

1-24-5-1004 Kagahara, Tsuzuki-ku, Yokohama-shi, Kanagawa-ken, Japan

Citizenship

Japan

Post Office Address

Same as above

Full name of third Inventor

Akira Koseki
Inventor's signature

Date

Residence

4-1-202 Matsugae-Chou, Sagamihara-shi, Kanagawa-ken

Citizenship

Japan

Post Office Address

Same as above

X This declaration ends with this page.